Lecture Notes on From Rules to Propositions

15-816: Linear Logic
Frank Pfenning

Lecture 2
January 18, 2012

We review the ideas of ephemeral truth and linear inference with another example from graph theory: constructing spanning trees for graphs. Then we probe the boundaries of what can be expressed in linear inference alone and start down the path of defining linear logical connectives in the form a *sequent calculus*. We also consider a first set of conditions to make sure the connectives are meaningful.

1 Example: Spanning Trees

A *spanning tree* for a connected graph is a graph that has the same nodes but only a subset of the edges such that there is no cycle. In order to define rules for constructing a spanning tree for a graph we will simultaneously manipulate two graphs: the original graph and its spanning tree. We therefore add a third argument to our representation of graphs (from Lecture 1) which identifies *which* graph a node or edge belongs to.

$\mathsf{node}(x, g)$	x is a node in graph g
$\mathsf{edge}(x, y, g)$	there is an edge from x to y in graph g

The rule of symmetry stays within one graph g:

$$\frac{\mathsf{edge}(x, y, g)}{\mathsf{edge}(y, x, g)}\ \mathsf{sym}$$

Now assume we have a graph g and want to build a spanning tree t. Here is a simple algorithm for building t. We begin by picking an arbitrary node

x from g and create t with x as its only node. Now we repeatedly pick an edge that connects a node x already in the tree with a node y not yet in the tree and add that edge and the node y into the tree. When no such edges exist any more, we either must have a spanning tree already or the original graph was not connected. We can determine this, for example, by checking if there are any nodes left in the graph that haven't been added to the tree.

This algorithm has two kinds of steps, so its representation in linear logic has two rules. The first step moves an arbitrary node from the graph to the tree.

$$\frac{\mathsf{node}(x, y, \mathsf{g})}{\mathsf{node}(x, y, \mathsf{t})}\ \mathsf{start?}$$

This rule can be used only once, at the very beginning of the algorithm and must be prohibited afterwards, or we could just use it to move all nodes from the graph to the tree without moving any edges. So we can either say the rule must be ephemeral itself, or we create a new ephemeral proposition init which only exists in the initial state and is consumed by the first step.

$$\frac{\mathsf{init} \quad \mathsf{node}(x, \mathsf{g})}{\mathsf{node}(x, \mathsf{t})}\ \mathsf{start}$$

The next rule implements the idea we described in the text above. All propositions are ephemeral, so we can implement "*a node y not yet in the tree*" by checking whether it is still in the graph, thereby consuming it.

$$\frac{\mathsf{node}(x, \mathsf{t}) \quad \mathsf{edge}(x, y, \mathsf{g}) \quad \mathsf{node}(y, \mathsf{g})}{\mathsf{node}(x, \mathsf{t}) \quad \mathsf{edge}(x, y, \mathsf{t}) \quad \mathsf{node}(y, \mathsf{t})}\ \mathsf{move}$$

A proof using these two rules describes a particular sequence of moves, taking edges from the graph and adding them to the spanning tree.

In order to convince ourselves that this is correct, it is important to understand the state invariants. Initially, we have

init	
$\mathsf{node}(x, \mathsf{g})$	for every node x in g
$\mathsf{edge}(x, y, \mathsf{g})$	for every edge from x to y in g

Rule move does not apply, because we do not yet have a node in t, so any inference must begin with rule start, consuming init and producing one node x_0 in t.

$\mathsf{node}(x_0, \mathsf{t})$	for some node x_0
$\mathsf{node}(x, \mathsf{g})$	for every node $x \neq x_0$ in g
$\mathsf{edge}(x, y, \mathsf{g})$	for every edge from x to y in g

Now rule start can no longer be applied, and we apply move as long as we can. The rule preserves the invariant that each node x from the initial graph is either in t (node(x, t)) or in g (node(x, g)). It further preserves the invariant that each edge in the original graph is either in t (edge(x, y, t)) or still in g (edge(x, y, g)).

If the algorithm stops and no nodes are left in g, we must have moved all n nodes originally in g. One is moved in the start rule, and $n - 1$ are moved in applications of the move rule. In every application of the move rule we also move exactly one edge from g to t, so t now has n nodes and $n - 1$ edges. Further, it is connected since anytime we move an edge it connects to something already in the partial spanning tree. A connected graph with n nodes and $n - 1$ edges must be a tree, and it spans g because it has all the nodes of g.

If the algorithms stops and there are some nodes left in g, then the original graph must have been disconnected. Assume that g is connected, y is left in g, and we started with x_0 in the first step. Because g is connected, there must be a path from x_0 to y. We prove that this is impossible by induction on the structure of this path. The last edge connects some node y' to y. If y' is in the tree, then the rule move would apply, but we stipulated that the algorithm only stops if move does not apply. If y' is in the graph but not in the tree, then we apply the induction hypothesis to the subpath from x_0 to y'.

2 Example: Beggars

The mechanism of inference we have introduced so far is quite elegant and already expressive. However, it has shortcomings. Notationally, it can be awkward to write down proofs. More importantly, there are some natural patterns of reasoning we would like to express and cannot because we have no logical connectives. Here is one example:

> *If wishes were horses, beggars would ride.* — English proverb

Let's start by analyzing "*if wishes were horses*". We can represent that by a (persistent) rule

$$\frac{\mathsf{wish}(x)}{\underline{\mathsf{horse}}(x)}$$

Here we have made wishes ephemeral and horses (or, perhaps, the intrinsic attribute for x to be a horse) persistent. This rule is schematic in x. Let's complete our vocabulary

$\mathsf{horse}(x)$	x is a horse
$\mathsf{wish}(x)$	x is a wish
$\mathsf{beggar}(x)$	x is a beggar
$\mathsf{rides}(x, y)$	x rides y

How would we say "*beggars would ride*"? We might make things more explicit by stating that for every y who is a beggar there is a horse z such that y rides z. Unfortunately, the whole statement requires "*wishes were horses*" to appear as a premise and "*beggers would ride*" as a conclusion, which is outside the scope of the inference rule notation. Making up something hypothetical (and allowing ourselves $\forall$ and $\exists$ quantifiers), it might look like:

$$\frac{\forall x \left(\dfrac{\mathsf{wish}(x)}{\underline{\mathsf{horse}}(x)} \right) \quad \underline{\mathsf{beggar}}(y)}{\exists z\,(\underline{\mathsf{horse}}(z) \quad \mathsf{rides}(y, z))}$$

Note that the universal quantifier is necessary in the premise. If we omitted it, then it would be enough to find one wish x that actually is a horse. Similarly, the existential quantifier in the conclusion is necessary, because if z were a schematic variable in the rule, then it could be instantiated arbitrarily, asserting that any begger could ride any arbitrary horse.

Analyzing this example we see that several key ingredients are necessary to express this are expressing inference rules as propositions, expressing persistence as an attribute of a proposition, and expressing quantifiers as propositions. Will will start in this lecture and complete the task in the next lecture.

3 Simultaneous Conjunction

Linear inference rules can have multiple premises and multiple conclusions. If we try to think of the horizontal line as some form of binary connective (it will turn out to be $A \multimap B$[1]), then we need a way to package up the premises to become a single proposition and the conclusions to become a single proposition. This is the purpose of the *simultaneous conjunction* or *multiplicative conjunction* $A \otimes B$. It is true if both A and B are true in the

[1] read as "*A linearly implies B*" or "*A lolli B*"

same state. So if we have $A \otimes B$ we can replace it by A and B:

$$\frac{A \otimes B \ \mathit{eph}}{A \ \mathit{eph} \quad B \ \mathit{eph}}$$

The other direction seems similarly straightforward: we can get $A \otimes B$ if we have both A and B:

$$\frac{A \ \mathit{eph} \quad B \ \mathit{eph}}{A \otimes B \ \mathit{eph}}$$

But this already creates problems. Say we want to show that

$$\frac{B \otimes A \ \mathit{eph}}{A \otimes B \ \mathit{eph}}$$

is a derived rule of inference. It would have to look something like

$$\cfrac{\cfrac{A \otimes B \ \mathit{eph}}{A \ \mathit{eph} \qquad B \ \mathit{eph}}}{B \otimes A \ \mathit{eph}}$$

One small irritation is that the premises of the last rule are in the wrong order. More significantly, however, $A \otimes B \ \mathit{eph}$ appears in the proof of both premises of the last rule. This would seem to constitute a violation of the very basis of linear inference: ephemeral facts are used only once!

One can try to devise criteria if a compact structure like the one above is in fact a valid proof for a derived rule of inference, but they are invariably quite complicated and don't scale well to all of linear logic. A more promising and general alternative is to change our notation for inference rather drastically, as we do in the next section.

4 Resources and Goals

We now move to a notation where the main judgment explicitly tracks all the ephemeral propositions we have used during inference. We write

$$\underbrace{A_1 \ \mathit{eph}, \ldots, A_n \ \mathit{eph}}_{\Delta} \vdash C \ \mathit{eph}$$

and refer to Δ as the *resources* and C as the goal we have to achieve. In order to prove this we have to use all the resources in Δ *exactly once* in the proof that we can achieve C. This is an example of a *sequent* from Gentzen's *sequent calculus* [Gen35], the seminal paper which started proof theory as a subject of study. Gentzen, however, had structural rules that allowed us to duplicate or erase assumptions, which are purposely omitted here.

With the sequent notation we can now write

$$\frac{\Delta \vdash A\ eph \quad \Delta' \vdash B\ eph}{\Delta, \Delta' \vdash A \otimes B\ eph} \otimes R$$

In the conclusion we combine the resources Δ needed to prove A with Δ' needed to prove B. No resource can be shared between these two subproofs, which would constitute a violation of the ephemeral nature of resources. On the other hand, the order of the resources does not matter, so allow them to be reordered freely.

This is an example of a *right rule* that shows how to prove a proposition (that is, achieve a goal). Conversely, we have to specify how to *use* a proposition. We do this with a *left rule* that breaks down one of the current set of resources. This is straightforward here.

$$\frac{\Delta, A\ eph, B\ eph \vdash C\ eph}{\Delta, A \otimes B\ eph \vdash C\ eph} \otimes L$$

We cannot invent such left and right rules for the connectives arbitrarily. In the end, we would like to have a system where the logical propositions have the expected meaning, both intuitively and formally. We explain some criteria we may apply in the next section.

5 Identity and Cut

Fundamentally, we need a balance between resources on the left and goals on the right. This balance is independent of the particular set of connectives we have—it should hold for arbitrary propositions.

The first, called *identity*, states that a resource A by itself should always be sufficient to achieve the goal A.

$$\frac{}{A\ eph \vdash A\ eph}\ \mathsf{id}_A$$

In this rule we have to be careful not to allow any additional unused resources, because their intepretation is tight: any resource must be used exactly once. We often note the proposition to which the rule is applied in a subscript of the rule name, because this information will be significant in our study of the sequent calculus.

The second, called *cut*, states the opposite: achieving a goal A licenses us to assume A as a resource.

$$\frac{\Delta \vdash A \ eph \quad \Delta', A \ eph \vdash C \ eph}{\Delta, \Delta' \vdash C \ eph} \ \mathsf{cut}_A$$

In this rule we have to be careful to combine the assumptions from both premises, again because all resources in the conclusion must be used exactly once (either in the proof of A or in the proof of C using A).

These two rules are sometimes called *judgmental rules*, because they are concerned with the nature of the judgments (here: of being a resource and a goal) or *structural rules*, because they do not examine the propositions but only the structure of the sequent.

In the rest of the lecture we will omit the judgment annotation eph since for now it is always the same.

6 Identity Expansion

Next we come to the criteria we apply to check that our definitions of connectives via the right- and left-rules are consistent with each other. The first checks that we can eliminate a use of the identity rule at a compound type to the uses of the identity rule at smaller types. This means that the left and right rules match well enough that we can derive the instances of the identity rule.

$$\frac{}{A \otimes B \vdash A \otimes B} \ \mathsf{id}_{A\otimes B} \quad \longrightarrow_E \quad \dfrac{\dfrac{\dfrac{}{A \vdash A}\ \mathsf{id}_A \quad \dfrac{}{B \vdash B}\ \mathsf{id}_B}{A, B \vdash A \otimes B}\ \otimes R}{A \otimes B \vdash A \otimes B}\ \otimes L$$

We write $\longrightarrow_E$ for the expansion an identity rule into a proof using identity rules at smaller propositions. This is called an expansion because the proof becomes larger, even if the propositions become smaller.

Note how we use the right and left rules for $A \otimes B$, and that they match up appropriately to derive the identity at $A \otimes B$.

7 Cut Reduction

The other judgmental rule is cut. It can also match up the right- and left rules for the same connective. This time we need to show that we can reduce a cut at $A \otimes B$ to cuts at A and B.

$$\dfrac{\dfrac{\Delta \vdash A \quad \Delta' \vdash B}{\Delta, \Delta' \vdash A \otimes B}\otimes R \quad \dfrac{\Delta'', A, B \vdash C}{\Delta'', A \otimes B \vdash C}\otimes L}{\Delta, \Delta', \Delta'' \vdash C}\,\mathsf{cut}_{A \otimes B}$$

$$\longrightarrow_R$$

$$\dfrac{\Delta' \vdash B \quad \dfrac{\Delta \vdash A \quad \Delta'', A, B \vdash C}{\Delta, \Delta'', B \vdash C}\,\mathsf{cut}_A}{\Delta, \Delta', \Delta'' \vdash C}\,\mathsf{cut}_B$$

Again, we see that the resources are in balance: we do not gain or lose any resources when we prove and then use $A \otimes B$.

Cut reduction turns out to be the engine behind some computational interpretations of linear logic, as we will see in later lectures.

8 Linear Implication

Finally, we return to the original problem of expressing inference rule. The idea is that a rule exchanging a quarter for two dimes and a nickel

$$\dfrac{\mathsf{q}}{\mathsf{d} \quad \mathsf{d} \quad \mathsf{n}}$$

becomes

$$\mathsf{q} \multimap \mathsf{d} \otimes \mathsf{d} \otimes \mathsf{n}$$

except that the rule itself would be persistent, while the proposition would not be, unless we make special arrangements.

Reading $A \multimap B$ aloud it means "*if we had an A we could obtain a B*". So, if we have $A \multimap B$ and we can obtain an A, that's license to use B.

$$\dfrac{\Delta \vdash A \quad \Delta', B \vdash C}{\Delta, \Delta', A \multimap B \vdash C}\multimap L$$

Conversely, to show that we can achieve $A \multimap B$ we have to show how to achieve B under the additional assumption that we have an A.

$$\frac{\Delta, A \vdash B}{\Delta \vdash A \multimap B} \multimap R$$

Let's check if they work together. First, identity expansion:

$$\frac{}{A \multimap B \vdash A \multimap B}\ \mathsf{id}_{A \multimap B} \quad \longrightarrow_E \quad \frac{\dfrac{\dfrac{}{A \vdash A}\ \mathsf{id}_A \quad \dfrac{}{B \vdash B}\ \mathsf{id}_B}{A \multimap B, A \vdash B} \multimap L}{A \multimap B \vdash A \multimap B} \multimap R$$

Next cut reduction:

$$\frac{\dfrac{\Delta, A \vdash B}{\Delta \vdash A \multimap B} \multimap R \quad \dfrac{\Delta' \vdash A \quad \Delta'', B \vdash C}{\Delta', \Delta'', A \multimap B \vdash C} \multimap L}{\Delta, \Delta', \Delta'' \vdash C}\ \mathsf{cut}_{A \multimap B}$$

$$\longrightarrow_R$$

$$\frac{\dfrac{\Delta' \vdash A \quad \Delta, A \vdash B}{\Delta, \Delta' \vdash B}\ \mathsf{cut}_A \quad \Delta'', B \vdash C}{\Delta, \Delta', \Delta'' \vdash C}\ \mathsf{cut}_B$$

Fortunately, the rules are once again in harmony.

We will continue with additional connectives and rules in the next lecture.

Exercises

Exercise 1 Consider variations of the representation and rules in the spanning tree example from Section 1. Consider all four possibilities of nodes and edges in g and t being ephemeral or persistent. In each case show the form of the three rules in question: sym (possibly with two variants), start, and move, indicate if the modification would be correct, and spell out how to check if a proper spanning tree has been built in the final state.

Exercise 2 We claimed that the rule

$$\frac{\mathsf{q}}{\mathsf{d} \quad \mathsf{d} \quad \mathsf{n}}$$

becomes

$$\mathsf{q} \multimap \mathsf{d} \otimes \mathsf{d} \otimes \mathsf{n}$$

Support this claim by proving that the following rules of inference are derived rules of the sequent calculus:

$$\frac{\Delta, \mathsf{d}, \mathsf{d}, \mathsf{n} \vdash C}{\Delta, \mathsf{q} \multimap \mathsf{d} \otimes \mathsf{d} \otimes \mathsf{n}, \mathsf{q} \vdash C} \qquad \frac{\Delta \vdash \mathsf{q} \quad \Delta', \mathsf{d}, \mathsf{d}, \mathsf{n} \vdash C}{\Delta, \Delta', \mathsf{q} \multimap \mathsf{d} \otimes \mathsf{d} \otimes \mathsf{n} \vdash C}$$

Exercise 3 An alternative left rule for linear implication would be

$$\frac{\Delta, B \vdash C}{\Delta, A, A \multimap B \vdash C} \multimap L'$$

Explore this rule

(i) Does it satisfy identity expansion?

(ii) Does it satisfy cut reduction?

(iii) If we have a system with the given rule $\multimap L$ can we derive $\multimap L'$ in it, and vice versa?

(iv) Discuss the relative merits of the two left rules, depending on the outcome of the above tests and other criteria you wish to apply.

Exercise 4 The 16-puzzle consists of a 4×4 square filled with 15 tiles, numbered 1 through 15, and one empty space. Tiles can slide horizontally or vertically into the empty space. For example, in the position on the left there are three possible moves, one of which transforms

1	2	3	4
5	6	7	8
9	10	11	12
13	14		15

into

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	

The goal is to achieve the situation depicted on the right, from some initial configuration.

We represent this with the following emphemeral predicates.

$\mathsf{sq}(x, y, n)$	Square (x, y) has tile n.
$\mathsf{empty}(x, y)$	Square (x, y) is empty.

The upper left-hand corner is square $(0, 0)$, the lower right-hand corner is $(\mathsf{s}(\mathsf{s}(\mathsf{s}(0))), \mathsf{s}(\mathsf{s}(\mathsf{s}(0))))$, with x increasing left-to-right and y increasing top-to-bottom. All numbers are represented in unary form, using 0 and $\mathsf{s}(-)$.

(i) Represent the legal moves. You may use a representation by logical formulas (in linear logic) or by inference rule with ephemeral premises and conclusions. Your rules should not depend on the size of the board.

(ii) Write rules to recognize the winning position for an arbitrary board with $c > 1$ columns and $r > 1$ rows. You should assume persistent facts

$$\underline{\mathsf{maxcol}}(c-1)$$
$$\underline{\mathsf{maxrow}}(r-1)$$
$$\underline{\mathsf{maxtile}}(c \times r - 1)$$

where $c-1$, $r-1$ and $c \times r - 1$ are precomputed in unary notation using 0 and $\mathsf{s}(-)$.

Your rules should have the property that a state $\Delta \vdash \mathsf{winning}$ if and only if Δ represents the winning configuration. If it is not winning, it may get stuck in an arbitrary state.

References

[Gen35] Gerhard Gentzen. Untersuchungen über das logische Schließen. *Mathematische Zeitschrift*, 39:176–210, 405–431, 1935. English translation in M. E. Szabo, editor, *The Collected Papers of Gerhard Gentzen*, pages 68–131, North-Holland, 1969.